Backpropagation through Time

(BPTT)
Backpropagation through time

- Input fed sequentially into network.

- Training is as if the network were unrolled to accommodate the entire sequence of input samples.

- Only one set of weights is actually used in operation; the weight changes are averaged across stages to get the actual weight change.
Backpropagation through time

\[ x_0, x_1, x_2, \ldots \rightarrow \text{Neural Net} \rightarrow y_0, y_1, y_2, \ldots \]

\[ \Delta W \rightarrow \text{average} \]

\[ x_0 \rightarrow y_0 \rightarrow x_1 \rightarrow y_1 \rightarrow x_2 \rightarrow y_2 \rightarrow \ldots \]

\[ \text{backpropagation} \]
BPTT Application

- The Truck Backer-Upper, D. Nguyen and B. Widrow
- Problem: Back up a truck so that 
  \[(x_{\text{trailer}}, y_{\text{trailer}}) = (x_{\text{dock}}, y_{\text{dock}}), \text{ given initial values for } (x_{\text{trailer}}, y_{\text{trailer}}, x_{\text{cab}}, y_{\text{cab}}, \theta_{\text{trailer}}, \theta_{\text{cab}})\]
Truck-Backer Problem

Figure 1: The truck, the trailer, and the loading dock
Training the Truck-Backer

- The truck moves in small time increments $\Delta$.

- A neural net is first trained to mimic the truck backing using **real truck dynamics**.

- Given the current state at a time $t$ (which includes the steering angle), the network learns to determine the next state (at time $t + \Delta$).

- This is done by starting the truck in a random state, observing the error between what the network does and the dynamic model, and adjusting the weights.
Kinematics as a Neural Net

Figure 2: Overview Diagram
Training Kinematics Network

Figure 3: Training the neural-net truck emulator
The function being learned

\[(x_{\text{trailer}}, y_{\text{trailer}}, x_{\text{cab}}, y_{\text{cab}}, \theta_{\text{trailer}}, \theta_{\text{cab}}) \xrightarrow{\theta_{\text{steering}}} \text{Truck} \xrightarrow{\text{}} (x'_{\text{trailer}}, y'_{\text{trailer}}, x'_{\text{cab}}, y'_{\text{cab}}, \theta'_{\text{trailer}}, \theta'_{\text{cab}})\]
Truck-Controller Combo

\[(x_{\text{trailer}}, y_{\text{trailer}}, x_{\text{cab}}, y_{\text{cab}}, \theta_{\text{trailer}}, \theta_{\text{cab}})\]

\[\rightarrow\]

Controller

\[\rightarrow\]

\[\theta_{\text{steering}}\]

Truck

\[\rightarrow\]

\[(x'_{\text{trailer}}, y'_{\text{trailer}}, x'_{\text{cab}}, y'_{\text{cab}}, \theta'_{\text{trailer}}, \theta'_{\text{cab}})\]

Already trained

to be trained
Training the Truck-Backer

- Starting from a random position, the controller backs up the truck one step at a time, until the goal is reached, or an obstacle (such as a side wall) is hit.

- An error value is produced by comparing the desired final state with the goal.

- The error value is backpropagated through the controller-truck combination to adjust the controller’s weights, using BPTT.
BPTT for truck training

\[ T = \text{Truck (already trained, weights fixed),} \]
\[ C = \text{Controller (being trained)} \]

\[ (x'_{\text{trailer}}, y'_{\text{trailer}}, x'_{\text{cab}}, y'_{\text{cab}}, \theta'_{\text{trailer}}, \theta'_{\text{cab}}) \]

state \(0\) state \(1\) state \(2\) ... state \(n-1\) state \(n\)

\[ \Sigma \]

\[ + \]

\[ - \]

\[ \theta_{\text{steering}} \]

error (for backprop)
Network Statistics

- Truck Emulator:
  - 6-45-6 tansig-tansig network

- Controller
  - 6-25-1 tansig-tansig network
Figure 6: Details of emulator and controller
Training

- 20,000 trials required to train
  - 16 lessons of 1000-2000 each

- Initially truck positioned very close to dock and in a nearly-correct position, so controller could learn easy tasks first.

- Final MSE was 3% of truck length, angle 7 degrees
Simulations

Figure 7
Simulations

- Initial state
- Time lapse
- Final state

dock
Simulations

initial state

time lapse

final state